

Draft Paper Mikrokontroller

Disusun guna memenuhi Evaluasi Akhir Semester mata kuliah Mikrokontroller

Dosen Pengampu:

Dr. Basuki Rahmat, S.Si. MT



Disusun oleh:

Miftahus Syifaul Haqqi

(23081010197)

PROGRAM STUDI INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS PEMBANGUNAN NASIONAL "VETERAN" JAWA TIMUR

TAHUN 2025

IMPLEMENTASI SISTEM PREDIKTIF PEMELIHARAAN MOTOR DC PADA KIT IMCLAB BERBASIS TINYML DAN PROTOKOL MQTT

Abstrak

Pemeliharaan motor DC merupakan aspek penting dalam sistem kendali dan otomasi, khususnya pada lingkungan pembelajaran berbasis praktikum. Kerusakan motor yang tidak terdeteksi sejak dini dapat menyebabkan penurunan performa hingga kegagalan sistem. Penelitian ini mengusulkan implementasi sistem prediktif pemeliharaan (predictive maintenance) motor DC pada Kit iMCLab dengan memanfaatkan teknologi TinyML dan protokol komunikasi MQTT. TinyML digunakan untuk menjalankan model machine learning berukuran ringan langsung pada perangkat mikrokontroler, sedangkan MQTT berperan sebagai protokol komunikasi untuk pengiriman data sensor secara real-time. Sistem dirancang untuk memantau parameter motor DC seperti kecepatan, arus, dan suhu, kemudian memprediksi kondisi motor ke dalam kategori normal atau berpotensi mengalami kerusakan. Hasil pengujian menunjukkan bahwa sistem mampu melakukan inferensi secara lokal pada perangkat dengan latensi rendah serta mengirimkan data dan hasil prediksi ke dashboard IoT secara andal. Implementasi ini membuktikan bahwa Kit iMCLab dapat digunakan sebagai media pembelajaran yang relevan untuk memahami konsep TinyML, IoT, dan predictive maintenance.

Kata kunci: Predictive Maintenance, Motor DC, TinyML, MQTT, iMCLab.

1. Pendahuluan

Motor DC merupakan salah satu komponen utama yang banyak digunakan dalam sistem kendali, robotika, otomasi industri, dan media pembelajaran berbasis embedded system. Pada lingkungan laboratorium pendidikan, motor DC sering dijadikan objek praktikum untuk memahami konsep dasar seperti pengaturan kecepatan, arah putaran, serta karakteristik torsi dan arus. Meskipun demikian, penggunaan motor DC secara terus-menerus dalam praktikum berpotensi menyebabkan penurunan performa hingga kerusakan komponen apabila tidak dilakukan pemantauan kondisi secara berkala.

Pendekatan pemeliharaan yang umum digunakan dalam kegiatan praktikum masih bersifat reaktif, yaitu perbaikan dilakukan setelah motor mengalami kegagalan fungsi. Pendekatan ini kurang efisien karena dapat mengganggu proses pembelajaran, meningkatkan biaya penggantian komponen, serta mengurangi umur pakai perangkat. Oleh karena itu, diperlukan pendekatan pemeliharaan yang lebih proaktif dan cerdas.

Seiring berkembangnya Artificial Intelligence (AI) dan Internet of Things (IoT), konsep predictive maintenance menjadi solusi yang banyak diterapkan pada sistem industri modern. Predictive maintenance memungkinkan sistem untuk memprediksi potensi kerusakan berdasarkan pola data sensor yang dikumpulkan secara kontinu. Namun, penerapan AI pada perangkat laboratorium pendidikan sering terkendala oleh keterbatasan sumber daya perangkat keras.

TinyML hadir sebagai pendekatan yang memungkinkan model machine learning berukuran kecil dijalankan langsung pada mikrokontroler dengan keterbatasan memori dan daya komputasi. Dengan TinyML, proses inferensi dapat dilakukan secara lokal (edge computing) tanpa bergantung sepenuhnya pada server atau cloud. Sementara itu, protokol MQTT digunakan untuk mendukung komunikasi data secara real-time dan ringan dalam sistem IoT.

Berdasarkan latar belakang tersebut, penelitian ini mengimplementasikan sistem prediktif pemeliharaan motor DC pada Kit iMCLab berbasis TinyML dan protokol MQTT. Tujuan penelitian ini adalah untuk merancang, mengimplementasikan, dan mengevaluasi sistem yang mampu memantau kondisi motor DC, melakukan prediksi kondisi secara lokal, serta menyajikan informasi hasil pemantauan melalui platform IoT.

2. Tinjauan Pustaka

2.1 Motor DC dan Pemeliharaan

Motor DC merupakan aktuator elektrome-kanik yang bekerja berdasarkan prinsip gaya Lorentz, di mana interaksi antara medan magnet dan arus listrik pada kumparan menghasilkan gaya putar pada rotor. Energi listrik yang diberikan pada motor dikonversi menjadi energi mekanik dalam bentuk torsi dan kecepatan sudut. Kinerja motor DC sangat dipengaruhi oleh beberapa parameter utama, yaitu tegangan, arus, kecepatan putar, dan suhu operasi.

Tegangan suplai berperan langsung dalam menentukan kecepatan putar motor. Penurunan atau fluktuasi tegangan yang tidak normal dapat menyebabkan motor bekerja di luar titik operasi optimal, sehingga berdampak pada penurunan efisiensi dan peningkatan rugi-rugi daya. Sementara itu, arus listrik mencerminkan beban kerja motor. Kenaikan arus yang signifikan sering kali mengindikasikan adanya peningkatan beban mekanik, gesekan berlebih, atau gangguan pada komponen internal motor, seperti bantalan atau kumparan.

Kecepatan putar motor merupakan parameter kinerja utama yang menggambarkan kemampuan motor dalam menghasilkan gerak mekanik sesuai dengan perintah sistem kendali. Penurunan kecepatan yang tidak sebanding dengan perubahan tegangan atau beban dapat menjadi indikasi awal terjadinya degradasi performa, seperti keausan sikat (brush), ketidakseimbangan rotor, atau kerusakan sebagian pada kumparan. Oleh karena itu, pemantauan kecepatan putar secara kontinu menjadi penting dalam mendeteksi anomali operasional.

Selain itu, suhu merupakan parameter kritis yang berkaitan erat dengan kondisi kesehatan motor DC. Peningkatan suhu yang berlebihan dapat disebabkan oleh arus berlebih, pendinginan yang tidak optimal, atau degradasi isolasi kumparan. Suhu operasi yang melebihi batas aman dalam jangka waktu tertentu dapat mempercepat penuaan material isolasi dan meningkatkan risiko kegagalan permanen pada motor.

Perubahan yang tidak normal pada satu atau lebih parameter tersebut umumnya tidak terjadi secara tiba-tiba, melainkan mengikuti pola tertentu seiring waktu. Pola perubahan inilah yang dapat dimanfaatkan sebagai indikator awal terjadinya kerusakan. Dengan memantau dan menganalisis parameter tegangan, arus, kecepatan, dan suhu secara simultan, sistem dapat mengidentifikasi kondisi motor sejak tahap awal degradasi performa. Pendekatan ini menjadi dasar dalam penerapan

predictive maintenance, di mana tindakan perawatan dapat dilakukan sebelum motor mengalami kegagalan total.

2.2 Predictive Maintenance

Predictive maintenance merupakan strategi pemeliharaan yang memanfaatkan data operasional dan teknik analisis untuk memprediksi kondisi peralatan di masa mendatang serta mengidentifikasi potensi kegagalan sebelum benar-benar terjadi. Pendekatan ini didasarkan pada pemantauan kondisi (condition-based monitoring), di mana data sensor dikumpulkan secara kontinu dari sistem yang diamati dan dianalisis untuk mendeteksi pola anomali atau kecenderungan penurunan kinerja.

Berbeda dengan pemeliharaan reaktif yang dilakukan setelah terjadinya kerusakan, predictive maintenance memungkinkan tindakan perawatan dilakukan secara proaktif pada waktu yang paling tepat. Selain itu, dibandingkan dengan pemeliharaan preventif yang mengandalkan jadwal tetap, predictive maintenance bersifat adaptif karena keputusan perawatan didasarkan pada kondisi aktual peralatan. Hal ini menjadikan predictive maintenance lebih efisien dalam hal penggunaan sumber daya dan waktu operasional.

Dalam konteks sistem motor listrik, predictive maintenance umumnya memanfaatkan parameter-parameter operasional seperti arus, tegangan, kecepatan, getaran, dan suhu sebagai indikator kondisi kesehatan motor. Perubahan pola pada parameter tersebut dapat mengindikasikan adanya degradasi komponen, peningkatan beban, atau gangguan pada sistem mekanik maupun elektrik. Dengan menganalisis data historis dan real-time, sistem dapat memprediksi kecenderungan kerusakan dan memberikan peringatan dini kepada pengguna.

Penerapan predictive maintenance terbukti mampu mengurangi downtime sistem, memperpanjang umur pakai peralatan, serta menekan biaya perawatan dan penggantian komponen. Dalam lingkungan pembelajaran dan laboratorium, pendekatan ini juga memberikan nilai tambah edukatif karena mahasiswa dapat mempelajari konsep pemeliharaan berbasis data dan kecerdasan buatan secara langsung melalui implementasi nyata.

2.3 TinyML

TinyML merupakan pendekatan machine learning yang dirancang untuk memungkinkan model berukuran kecil dijalankan secara langsung pada perangkat embedded yang memiliki keterbatasan sumber daya, baik dari sisi memori, daya komputasi, maupun konsumsi energi. Berbeda dengan pendekatan machine learning konvensional yang umumnya membutuhkan komputasi tinggi dan dijalankan pada server atau cloud, TinyML menekankan pada efisiensi model sehingga proses inferensi dapat dilakukan pada perangkat edge secara mandiri.

Karakteristik utama TinyML meliputi ukuran model yang kecil, latensi inferensi yang rendah, serta konsumsi daya yang minimal. Model TinyML biasanya dioptimalkan melalui pemilihan arsitektur sederhana, penggunaan jumlah parameter yang terbatas, serta teknik kuantisasi untuk mengurangi kebutuhan memori. Dengan pendekatan ini, model machine learning dapat dijalankan pada mikrokontroler dengan kapasitas memori yang terbatas tanpa mengganggu kinerja sistem utama.

Dalam konteks Kit iMCLab, penggunaan TinyML menjadi sangat relevan karena platform ini umumnya berbasis mikrokontroler yang dirancang untuk kegiatan praktikum dan pembelajaran. Integrasi TinyML memungkinkan Kit iMCLab tidak hanya berfungsi sebagai alat kendali dan monitoring, tetapi juga sebagai sistem cerdas yang mampu melakukan analisis data dan pengambilan keputusan secara lokal. Hal ini mendukung penerapan konsep edge computing, di mana pemrosesan data dilakukan sedekat mungkin dengan sumber data.

Penerapan TinyML pada sistem pemeliharaan prediktif motor DC memungkinkan proses klasifikasi atau prediksi kondisi motor dilakukan secara real-time tanpa ketergantungan pada koneksi internet yang stabil. Data sensor seperti arus, kecepatan, dan suhu dapat diproses langsung pada mikrokontroler untuk menentukan kondisi kesehatan motor. Pendekatan ini meningkatkan keandalan sistem, mengurangi latensi komunikasi, serta menjadikan sistem lebih efisien dari sisi energi dan bandwidth jaringan.

Selain manfaat teknis, penggunaan TinyML pada Kit iMCLab juga memiliki nilai edukatif yang tinggi. Mahasiswa dapat mempelajari secara langsung bagaimana model machine learning dirancang, dioptimalkan, dan diimplementasikan pada perangkat dengan sumber daya terbatas. Dengan demikian, TinyML menjadi jembatan antara konsep kecerdasan buatan dan implementasi nyata pada sistem embedded dan IoT.

2.4 Protokol MQTT

MQTT (Message Queuing Telemetry Transport) merupakan protokol komunikasi ringan yang dirancang khusus untuk sistem terdistribusi dengan keterbatasan sumber daya, seperti perangkat IoT dan sistem embedded. MQTT menggunakan arsitektur *publish-subscribe*, di mana perangkat pengirim data (*publisher*) dan penerima data (*subscriber*) tidak berkomunikasi secara langsung, melainkan melalui sebuah perantara yang disebut *broker*. Arsitektur ini memungkinkan sistem menjadi lebih fleksibel, skalabel, dan mudah dikembangkan.

Keunggulan utama MQTT terletak pada overhead komunikasi yang rendah serta kebutuhan bandwidth yang minimal. Hal ini menjadikan MQTT sangat sesuai untuk aplikasi yang memerlukan pengiriman data secara real-time namun berjalan pada jaringan dengan keterbatasan kapasitas atau konektivitas yang tidak stabil. Selain itu, MQTT mendukung mekanisme *Quality of Service* (QoS) yang memungkinkan pengaturan tingkat keandalan pengiriman pesan sesuai dengan kebutuhan aplikasi, mulai dari pengiriman tanpa jaminan hingga pengiriman yang terjamin sampai ke penerima.

Dalam sistem IoT, MQTT banyak digunakan untuk pemantauan dan kendali perangkat secara jarak jauh karena kemampuannya dalam menangani komunikasi data secara efisien dan asinkron. Data sensor dapat dikirimkan secara periodik atau berdasarkan peristiwa tertentu tanpa membebani jaringan secara berlebihan. Karakteristik ini sangat penting dalam sistem yang melibatkan banyak node IoT yang beroperasi secara bersamaan.

Pada implementasi sistem prediktif pemeliharaan motor DC pada Kit iMCLab, MQTT berperan sebagai media komunikasi utama untuk mengirimkan data sensor dan hasil inferensi dari model TinyML. Mikrokontroler berfungsi sebagai *publisher* yang mengirimkan informasi kondisi motor

ke broker MQTT, sementara dashboard atau aplikasi monitoring bertindak sebagai *subscriber* yang menerima dan menampilkan data tersebut secara real-time. Dengan pendekatan ini, pemantauan kondisi motor dapat dilakukan secara jarak jauh tanpa mengganggu proses inferensi lokal pada perangkat.

Integrasi MQTT dengan TinyML juga mendukung konsep edge–cloud collaboration, di mana pemrosesan utama dilakukan pada perangkat edge, sementara MQTT digunakan untuk distribusi data dan visualisasi. Pendekatan ini meningkatkan efisiensi sistem, mengurangi latensi komunikasi, serta menjadikan arsitektur sistem lebih andal dan mudah diperluas.

3. Metodologi Penelitian

Metodologi penelitian yang digunakan dalam penelitian ini bersifat eksperimental dan terdiri dari beberapa tahapan sistematis untuk memastikan sistem yang dikembangkan berjalan sesuai dengan tujuan yang diharapkan.

3.1 Akuisisi Data Sensor

Tahap awal penelitian adalah pengambilan data dari motor DC yang terpasang pada Kit iMCLab. Data diperoleh dari beberapa sensor, antara lain sensor kecepatan putar motor, sensor arus, dan sensor suhu. Data dikumpulkan dalam berbagai kondisi operasi, seperti beban ringan, beban berat, serta kondisi mendekati kegagalan.

3.2 Pra-pemrosesan dan Pelabelan Data

Data sensor yang telah dikumpulkan kemudian diproses untuk menghilangkan noise dan menyesuaikan skala data. Selanjutnya, data diberi label kondisi motor, misalnya kondisi normal dan kondisi tidak normal. Proses pelabelan ini menjadi dasar dalam pelatihan model machine learning.

3.3 Perancangan dan Pelatihan Model TinyML

Model machine learning dirancang menggunakan algoritma yang ringan dan sesuai untuk perangkat embedded, seperti logistic regression atau neural network sederhana. Model dilatih menggunakan dataset yang telah diproses, kemudian dievaluasi akurasi. Setelah itu, model dikonversi ke format TinyML agar dapat dijalankan pada mikrokontroler.

3.4 Integrasi Sistem IoT Berbasis MQTT

Pada tahap ini, sistem komunikasi IoT diintegrasikan menggunakan protokol MQTT. Mikrokontroler berperan sebagai publisher yang mengirimkan data sensor dan hasil prediksi ke broker MQTT. Dashboard IoT berperan sebagai subscriber yang menerima dan menampilkan data secara real-time.

3.5 Pengujian dan Evaluasi Sistem

Pengujian dilakukan untuk mengevaluasi kinerja sistem secara keseluruhan. Parameter yang diuji meliputi akurasi prediksi model TinyML, latensi inferensi, penggunaan memori mikrokontroler,

serta kestabilan komunikasi MQTT. Hasil pengujian dianalisis untuk mengetahui keandalan sistem.

4. Implementasi Sistem

4.1 Arsitektur Sistem

Sistem yang dikembangkan terdiri dari beberapa komponen utama yang saling terintegrasi, yaitu motor DC sebagai objek yang dipantau, sensor sebagai sumber data, mikrokontroler sebagai node TinyML, broker MQTT sebagai perantara komunikasi, serta dashboard monitoring sebagai antarmuka pengguna. Setiap komponen memiliki peran spesifik dalam mendukung implementasi sistem prediktif pemeliharaan secara menyeluruh.

Motor DC berfungsi sebagai aktuator yang diamati kondisi operasionalnya. Untuk memperoleh informasi kondisi motor secara real-time, sistem dilengkapi dengan berbagai sensor, seperti sensor kecepatan putar, sensor arus, dan sensor suhu. Sensor-sensor tersebut menghasilkan data mentah yang merepresentasikan keadaan aktual motor selama beroperasi. Data ini kemudian dikirimkan ke mikrokontroler untuk diproses lebih lanjut.

Mikrokontroler berperan sebagai pusat pemrosesan sistem sekaligus sebagai node TinyML. Pada komponen ini, data sensor yang masuk terlebih dahulu melalui tahap pra-pemrosesan sederhana sebelum digunakan sebagai input model machine learning. Model TinyML yang telah dilatih sebelumnya disimpan di dalam memori mikrokontroler dan digunakan untuk melakukan inferensi secara lokal. Proses inferensi ini bertujuan untuk mengklasifikasikan atau memprediksi kondisi kesehatan motor DC, misalnya dalam kategori normal atau berpotensi mengalami kerusakan.

Inferensi yang dilakukan secara lokal pada mikrokontroler memungkinkan sistem bekerja secara mandiri tanpa ketergantungan penuh pada koneksi internet atau komputasi cloud. Pendekatan ini mengurangi latensi, meningkatkan keandalan sistem, serta menekan penggunaan bandwidth jaringan. Selain itu, inferensi lokal mendukung konsep edge computing yang sangat sesuai untuk aplikasi IoT berbasis perangkat embedded.

Hasil inferensi serta data sensor selanjutnya dikirimkan ke broker MQTT menggunakan mekanisme publish–subscribe. Mikrokontroler bertindak sebagai *publisher* yang mengirimkan data ke topik tertentu, sementara dashboard monitoring berperan sebagai *subscriber* yang menerima data tersebut. Dashboard monitoring digunakan untuk menampilkan informasi kondisi motor DC secara real-time, termasuk parameter sensor dan hasil prediksi kondisi motor, sehingga pengguna dapat melakukan pemantauan dan evaluasi sistem dengan lebih mudah.

Dengan arsitektur sistem ini, integrasi antara TinyML dan MQTT dapat diwujudkan secara efektif, di mana pemrosesan cerdas dilakukan pada sisi edge, sedangkan MQTT digunakan sebagai sarana distribusi informasi dan visualisasi data. Pendekatan ini menjadikan sistem prediktif pemeliharaan motor DC pada Kit iMCLab bersifat efisien, skalabel, dan relevan untuk diterapkan sebagai media pembelajaran berbasis AI dan IoT.

4.2 Implementasi TinyML

Model TinyML digunakan untuk melakukan proses klasifikasi kondisi motor DC berdasarkan data input yang diperoleh dari sensor. Data sensor tersebut mencakup parameter-parameter operasional utama, seperti arus listrik, kecepatan putar, dan suhu motor. Parameter-parameter ini dipilih karena memiliki korelasi langsung dengan kondisi kesehatan dan kinerja motor DC. Sebelum digunakan sebagai input model, data sensor terlebih dahulu melalui tahap pra-pemrosesan untuk memastikan kesesuaian format dan skala data dengan kebutuhan model TinyML.

Model TinyML yang diterapkan dirancang dengan arsitektur sederhana namun efektif agar dapat dijalankan pada mikrokontroler dengan keterbatasan memori dan daya komputasi. Proses inferensi dilakukan secara lokal pada perangkat, sehingga sistem mampu menghasilkan prediksi kondisi motor secara real-time. Output dari model berupa hasil klasifikasi yang menunjukkan status kondisi motor, misalnya dalam kategori normal atau tidak normal, yang merepresentasikan tingkat kesehatan motor DC.

Hasil klasifikasi yang diperoleh dari model TinyML digunakan sebagai dasar dalam penentuan status kesehatan motor. Status ini kemudian dimanfaatkan untuk memberikan indikasi dini terhadap potensi terjadinya kerusakan atau penurunan performa. Dengan adanya klasifikasi kondisi secara kontinu, sistem dapat mendeteksi perubahan pola operasional motor sejak tahap awal degradasi, sehingga tindakan pemeliharaan dapat direncanakan sebelum terjadi kegagalan total.

Pendekatan klasifikasi berbasis TinyML ini mendukung konsep predictive maintenance, di mana keputusan pemeliharaan didasarkan pada kondisi aktual peralatan, bukan hanya pada jadwal waktu tertentu. Selain meningkatkan keandalan sistem, pendekatan ini juga memberikan efisiensi dari sisi operasional dan edukatif. Dalam konteks Kit iMCLab, implementasi model TinyML memungkinkan mahasiswa untuk memahami secara langsung bagaimana data sensor diolah menjadi informasi kesehatan motor melalui algoritma machine learning yang dijalankan pada perangkat embedded.

4.3 Implementasi MQTT

MQTT digunakan sebagai protokol komunikasi untuk mengirimkan data sensor dan hasil prediksi kondisi motor DC ke dashboard monitoring secara real-time. Pada sistem ini, mikrokontroler berperan sebagai *publisher* yang secara periodik mengirimkan data parameter operasional motor, seperti arus, kecepatan, dan suhu, serta hasil inferensi dari model TinyML ke broker MQTT melalui topik tertentu. Mekanisme ini memungkinkan distribusi data dilakukan secara efisien tanpa membebani jaringan komunikasi.

Dashboard monitoring berfungsi sebagai *subscriber* yang menerima data dari broker MQTT dan menampilkan informasi kondisi motor secara visual kepada pengguna. Informasi yang ditampilkan mencakup nilai parameter sensor secara real-time serta status kesehatan motor berdasarkan hasil klasifikasi TinyML. Dengan demikian, pengguna dapat melakukan pemantauan kondisi motor dari jarak jauh tanpa harus berada langsung di lokasi perangkat.

Penggunaan MQTT mendukung komunikasi data dengan latensi rendah dan kebutuhan bandwidth yang minimal, sehingga sangat sesuai untuk aplikasi monitoring berbasis IoT. Selain itu, arsitektur

publish-subscribe pada MQTT memungkinkan sistem untuk dikembangkan secara fleksibel, misalnya dengan menambahkan lebih dari satu dashboard atau sistem analisis tanpa perlu mengubah konfigurasi pada perangkat mikrokontroler.

Integrasi MQTT dengan sistem TinyML juga memperkuat konsep pemrosesan terdistribusi, di mana inferensi dilakukan secara lokal pada perangkat embedded, sementara MQTT digunakan untuk penyebaran informasi dan visualisasi data. Pendekatan ini meningkatkan efisiensi sistem secara keseluruhan, mengurangi ketergantungan pada koneksi internet yang stabil, serta menjadikan sistem monitoring lebih andal dan responsif. Dengan kemampuan pemantauan jarak jauh ini, sistem prediktif pemeliharaan motor DC pada Kit iMCLab dapat digunakan secara efektif baik untuk keperluan praktikum maupun demonstrasi penerapan AI dan IoT.

5. Hasil dan Pembahasan

Hasil implementasi menunjukkan bahwa sistem prediktif pemeliharaan motor DC pada Kit iMCLab dapat berjalan sesuai dengan perancangan. Model TinyML berhasil dijalankan pada mikrokontroler dengan penggunaan memori yang relatif kecil dan waktu inferensi yang cepat, sehingga cocok untuk aplikasi real-time.

Data hasil pengujian memperlihatkan bahwa sistem mampu membedakan kondisi motor normal dan tidak normal berdasarkan pola data sensor. Pada kondisi beban berat dan suhu yang meningkat, sistem memberikan indikasi potensi penurunan performa motor. Hal ini menunjukkan bahwa pendekatan predictive maintenance dapat diterapkan secara efektif pada skala laboratorium pendidikan.

Dari sisi komunikasi, protokol MQTT terbukti mampu mengirimkan data sensor dan hasil prediksi secara stabil dengan latensi rendah. Dashboard IoT memungkinkan pengguna untuk memantau kondisi motor secara jarak jauh, sehingga proses praktikum menjadi lebih interaktif dan informatif.

Secara keseluruhan, integrasi TinyML dan MQTT pada Kit iMCLab memberikan pengalaman pembelajaran yang lebih komprehensif, karena mahasiswa tidak hanya mempelajari aspek kendali motor, tetapi juga konsep AI, IoT, dan edge computing secara langsung.

6. Kesimpulan

Penelitian ini berhasil mengimplementasikan sistem prediktif pemeliharaan motor DC pada Kit iMCLab berbasis TinyML dan MQTT. Sistem yang dikembangkan mampu melakukan prediksi kondisi motor secara lokal dan mengirimkan informasi secara real-time. Implementasi ini membuktikan bahwa Kit iMCLab dapat digunakan sebagai media pembelajaran yang aplikatif untuk memahami konsep AI dan IoT pada sistem embedded.

Daftar Pustaka

1. Banbury, C. R., et al. (2021). TinyML: Enabling On-Device Machine Learning. *Proceedings of MLSys*.
2. Lee, J., et al. (2014). Predictive Maintenance of Industrial Equipment. *International Journal of Advanced Manufacturing Technology*.
3. MQTT Version 3.1.1 Specification.
4. Dokumentasi dan Modul Praktikum iMCLab.